# Deep Learning Analysis for Alphabet Soup Charity Funding Predictions

Christina Land

February 27, 2025

Challenge 21

# Results

The preprocessing phase focused on transforming the raw dataset into a suitable format for the neural network.   
The target variable for this model was "IS\_SUCCESSFUL," which indicates whether an organization effectively utilized its funding.   
The features included application type, affiliation, classification, use case, organization, status, income amount, special considerations,   
and requested funding amount. Two columns, "EIN" and "NAME," were removed as they served as unique identifiers and did not contribute to   
predictive performance. Since some categorical variables had a large number of unique values, a data transformation step was applied to simplify modeling.   
Rare categories in the "APPLICATION\_TYPE" and "CLASSIFICATION" fields, which had fewer than 500 occurrences, were grouped into an "Other" category.   
This ensured that the model focused on the most significant classifications while avoiding sparsely represented data points.   
Next, categorical variables were encoded using one-hot encoding to convert them into numerical format.   
After encoding, the dataset was split into training and testing subsets, with 80% of the data used for training and 20% reserved for testing.   
To ensure that all input features were on the same scale, a StandardScaler was applied to normalize numerical variables.  
  
The initial deep learning model was built using a feedforward neural network with an input layer containing 80 neurons and a ReLU activation function.   
A hidden layer with 30 neurons, also utilizing ReLU activation, was added to improve the model’s ability to capture complex patterns.   
Finally, the output layer consisted of a single neuron with a sigmoid activation function to classify the funding success as either 1 (successful) or 0 (unsuccessful).   
The model was compiled using the Adam optimizer and binary cross-entropy as the loss function, and it was trained for 50 epochs with a batch size of 32.   
The model’s performance showed a final training accuracy of approximately 73.9% and a test accuracy of 72.5%, with a corresponding loss of 0.5590.  
  
Since the initial model's accuracy did not meet the 75% threshold, multiple optimization strategies were implemented to improve its performance.   
The first optimization attempt involved increasing the number of neurons and adding an additional hidden layer, adjusting the architecture to 100, 50, and 25 neurons per layer.   
This modification resulted in minor accuracy improvements but did not significantly enhance model performance.   
The second approach focused on modifying activation functions by replacing ReLU with Leaky ReLU and incorporating dropout layers with a probability of 0.2 to prevent overfitting.   
This optimization enhanced the model’s generalization but still fell short of the 75% accuracy target.   
The third optimization strategy involved increasing the number of epochs to 100 while reducing the batch size to 16.   
This allowed for more frequent updates to the model's weights, leading to a more stable learning process.   
Despite these modifications, the model remained slightly below the desired accuracy threshold, indicating that deep learning may not be the most effective approach for this dataset.  
  
In summary, the deep learning model successfully identified patterns within the data and achieved an accuracy of approximately 73-74%.   
Although performance improved with optimization attempts, the model did not consistently reach the 75% benchmark.   
Given the structured nature of the dataset, a tree-based machine learning model such as Random Forest or XGBoost may yield better results.   
These models handle categorical variables efficiently without extensive one-hot encoding and provide feature importance metrics, which improve interpretability.   
Additionally, ensemble learning techniques or further hyperparameter tuning could enhance prediction accuracy.